Universitat Autònoma de Barcelona

Facultat de Ciències

Pràctica Neo4j

Bases de dades no relacionals

Sergi Cantón Simó - 1569251

Bernat Espinet Torrescassana - 1564342

Marc Llopart Enajas - 1569054

Gerard Vinyes Sanchez - 1563545

10/6/2022

Índex

1 Repositori GitHub

Per fer la pràctica, hem creat un repositori GitHub per a emmagatzemar i anar actualitzant els fitxers que hem anat fer servir.

L'enllaç al repositori és https://github.com/MarcLlopart/BDnR_MATCAD_Grup5.

En aquest repositori podem trobar cinc arxius. El primer, *DADES*, és una carpeta que conté les dades necessàries per dur a terme les consultes del projecte. El segon fitxer, *Informe.pdf* és aquest mateix informe, on es troben les explicacions i justificacions de la pràctica. En el tercer fitxer, *Exercici1.txt* hi ha el codi que llegeix les dades i crea els nodes i relacions corresponents per generar el graf de la pràctica. Al quart fitxer, *Exercici2.txt*, hi ha el codi de les consultes que se'ns demanaven a l'exercici 2. Finalment, *Exercici3.txt*, correspon al codi de l'exercici 3, on es duu a terme un procés d'analítica del graf que hem utilitzat.

2 Exercicis

2.1 Exercici 1. Importació de les dades

El primer exercici és fonamental per poder continuar amb la pràctica. Aquí se'ns demanava implementar el codi necessari per a importar les dades al Neo4j. Això ho vàrem fer dins del fitxer *Exercici1_Neo4j_BNR.txt*.

Partim de cinc fitxers csv: *FAMILIA.csv*, *HABITATGES.csv*, *INDIVIDUAL.csv*, *SAME_AS.csv* i *VIU.csv*. Aquests contenen la informació de les dades dels nodes i de les relacions.

Dins del fitxer *Exercici1_Neo4j_BNR.txt*, inicialment s'esborren totes les dades. A continuació, s'eliminen també les restriccions d'unicitat. Posteriorment, es llegeixen les dades, es creen els nodes i relacions i es creen les restriccions necessàries. Cada fitxer csv es llegeix de manera diferent segons si estem executant el codi el local o en una màquina virtual d'Azure. En el primer cas, únicament cal posar el path al csv. En el segon cas, s'ha de posar un enllaç als fitxers en Google Drive.

El primer fitxer a tractar és el que conté les dades d'habitatges. Cada habitatge representa un node, que té els atributs municipi, *id_llar*, *any_padro*, carrer i *numero*. Després d'haver llegit les dades i haver creat els nodes corresponents a cada conjunt d'atributs, es crea la restricció d'unicitat dels habitatges. Com el nom diu, aquesta evita que hi pugui haver habitatges iguals repetits.

Seguidament, es tracta el fitxer d'individus. Cada node individu està format pels atributs *id*, *year*, *name* i *surname*, *second_surname*. D'igual manera que amb els habitatges, cal crear una restricció que asseguri que els individus no estiguin repetits.

Després d'haver creat els nodes, la primera relació que es tracta és la de vivenda. A partir del fitxer *VIU.csv*, es relaciona cada individu amb l'habitatge on viu. La segona és la de les relacions de parentesc. Aquí es relaciona cada persona amb els seus familiars. A la relació se li assigna l'atribut *relacio*, que determina el tipus de relació de parentesc d'una persona a l'altra. Finalment, la tercera és aquella que relaciona persones amb sí mateixes en diversos moments temporals, és a dir, nodes individu amb nodes individu.

2.1.1 Implementació al cloud

Com ja hem mencionat, existeix la possibilitat de treballar en remot, en una màquina virtual d'Azure. Per això, hem carregat tota la base de dades en un projecte de Neo4j al qual li hem assignat com a contrasenya *grup5*. A l'escriptori hem deixat una carpeta amb la informació de les consultes de l'exercici 2 i l'exercici 3.

2.2 Exercici 2. Consultes

El segon exercici consisteix en dur a terme una sèrie de consultes de la base de dades en cypher.

2.2.1 *Dels padró de 1866 de Castellví de Rosanes (CR), retorna el número d'habitants i la llista de noms. Elimina duplicats i nan.*

Resolem la consulta filtrant per les persones que viuen a Castellví de Rosanes i que el seu nom no sigui nan. Retornem el recompte de noms, és a dir, de persones, i una llista amb els diferents noms. La consulta és

MATCH (i:Individual)-[:VIU]->(h:Habitatge)

WHERE (i.name <> 'nan' and h.municipi='CR')

RETURN count(i.name) AS 'Num Habitants', collect(distinct i.name) AS 'Llistat';

i dona com a resultat

"Num Habitants"	"Llistat"
333	["jose","rosa","antonio","elisa","manuel","miguel","emilia","juan","magdalena","teresa","pedro","francisca","salvador","francisco","luis","isidro","paula","jaime","madrona","jacoba","joaquina","ramon","maria","consepcion","eulalia","mercedes","benito","pascuala","antinio","esteban","josefa","dolores","martin","carmen","antonia","isabel","angela","lorenzo","celestino","agustin","jacinto","maria angela","pablo","vicente","serafin","margarita","tomas","filomena","amelino","vicenta","catalina","esteban","sebastia","ignes","joaquin","balbina","jacinta","ilegible","maria rosa","florencia","mariangela","concepcion","rosendo","ines","merced","bartolome","magin","rosalia","camila","fran","lucia","cristobal","clemente"]

Figura 1: Resultat de la primera consulta.

2.2.2 *Dels padrons de Sant Feliu de Llobregat (SFLL) d'abans de l'any 1840 (no inclòs), retorna la població, l'any del padró i la llista d'identificadors dels habitatges de cada padró. Ordena els resultats per l'any de padró.*

Resolem la consulta filtrant pels habitatges de Sant Feliu de Llobregat l'any del qual sigui menor a 1840. Després, retornem els atributs de població, any i una llista, que retornem ordenada segons l'any, de tal manera que tota la taula queda ordenada de l' *id* més vell al més actual. La consulta és

MATCH (i:Individual)-[:VIU]-¿(h:Habitatge)

WHERE (h.municipi = 'SFLL' and h.any_padro ¡ 1840)

RETURN DISTINCT h.municipi AS 'Població',h.any_padro AS 'Any', collect(distinct(h.id_llar)) AS 'Llista Llars'

ORDER BY h.any_padro;

i dona com a resultat

"Població"	"Any"	"Llista Llars"
"SFLL"	1833	[10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32, 33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,5 6,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79 ,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,1 02,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119 ,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,1 37,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154 ,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,1 72,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189 ,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,2 07,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224 ,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,2 42,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259 ,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,2 77,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294 ,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,3 12,313]
"SFLL"	1838	[314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,3

Figura 2: Resultat de la segona consulta.

2.2.3 *Retorna el nom de les persones que vivien al mateix habitatge que "rafel marti" (no té segon cognom) segons el padró de 1838 de Sant Feliu de Llobregat (SFLL). Retorna la informació en mode graf i mode llista.*

Resolem la consulta trobant els nodes que estiguin relacionats amb la relació viu amb "rafel marti" i filtrant per aquells de l'any 1838 i del municipi Sant Feliu de Llobregat.

La consulta, si ho volem retornar en mode graf, és

MATCH (i1:Individual {name:"rafel", surname:"marti"})-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individual)

WHERE h.any_padro = 1838 AND h.municipi = "SFLL"

RETURN i1, i2;

i dona com a resultat

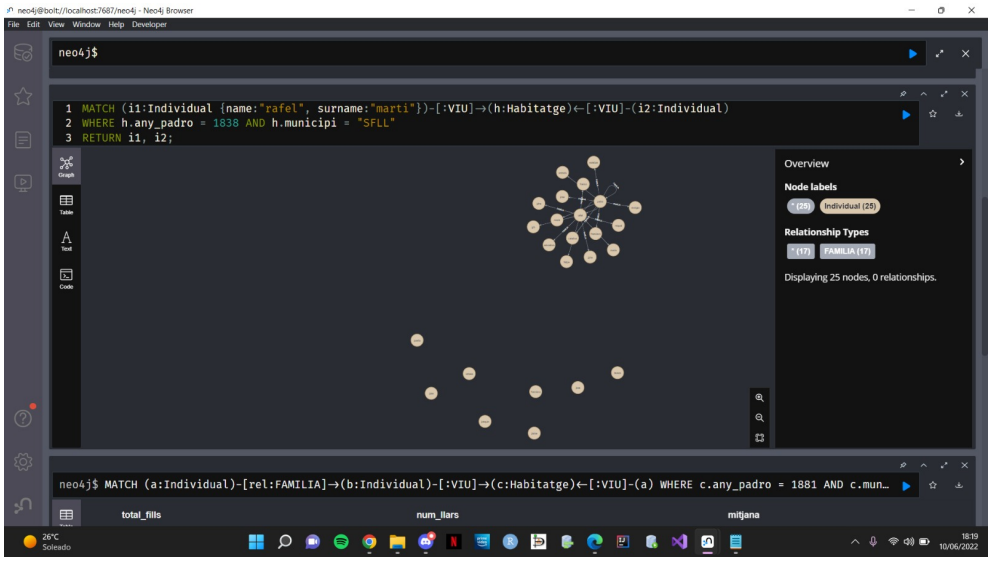


Figura 3: Resultat de la tercera consulta en format graf.

La consulta, si ho volem retornar en mode llista, és

MATCH (i1:Individual {name:"rafel", surname:"marti"})-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individual)

WHERE h.any_padro = 1838 AND h.municipi = "SFLL"

RETURN i1.name as 'Nom', collect(DISTINCT i2.name)AS 'Convivents';

i dona com a resultat

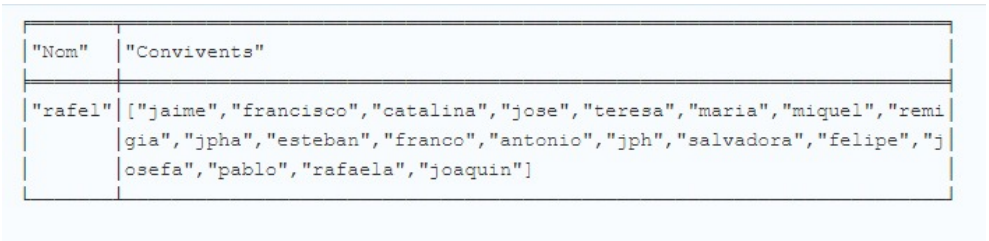


Figura 4: Resultat de la tercera consulta en format llista.

Altrament, si només volguéssim veure les persones que conviuen en el mateix habitatge i són família de "rafel marti", només hauríem d'afegir la relació de família entre un individu i l'altre, resultant en el següent *match*.

MATCH (i1:Individual {name:"rafel", surname:"marti"})-[:VIU]->(h:Habitatge)<-[:VIU]-(i2:Individual), (i1)-[:FAMILIA]->(i2)

2.2.4 *Retorna totes les aparicions de "Miguel ballester". Fes servir la relació SAME_AS per poder retornar totes les instancies, independentment de si hi ha variacions lèxiques (ex. diferents formes d'escriure el seu nom/cognoms). Mostra la informació en forma de subgraf.*

Resolem la consulta guardant-nos en forma de subgraf les relacions entre mateixes persones i filtrem per aquelles que corresponen a "miguel ballester". La consulta és

MATCH path=((i1:Individual)-[:SAME_AS]-(i2:Individual))

WHERE i1.name='miguel' and i1.surname='ballester'

return path;

i dona com a resultat

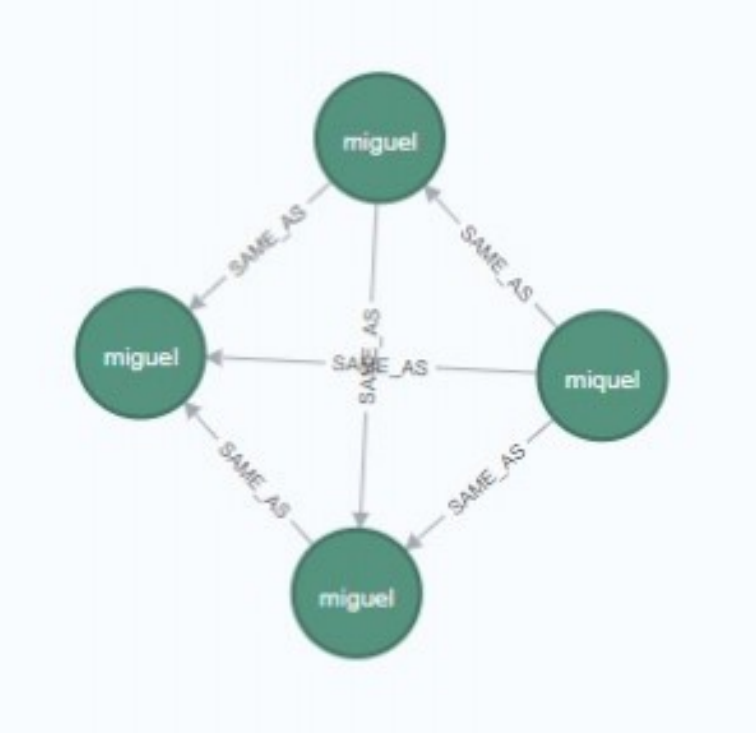


Figura 5: Resultat de la quarta consulta.

2.2.5 *5. Mostra totes les persones relacionades amb "antonio farran". Mostra la informació en forma de taula: el nom, cognom1, cognom2, i tipus de relació.*

Resolem la consulta buscant les persones relacionades de qualsevol manera amb "antonio farran" i retornem els atributs que se'ns demanen. La consulta és

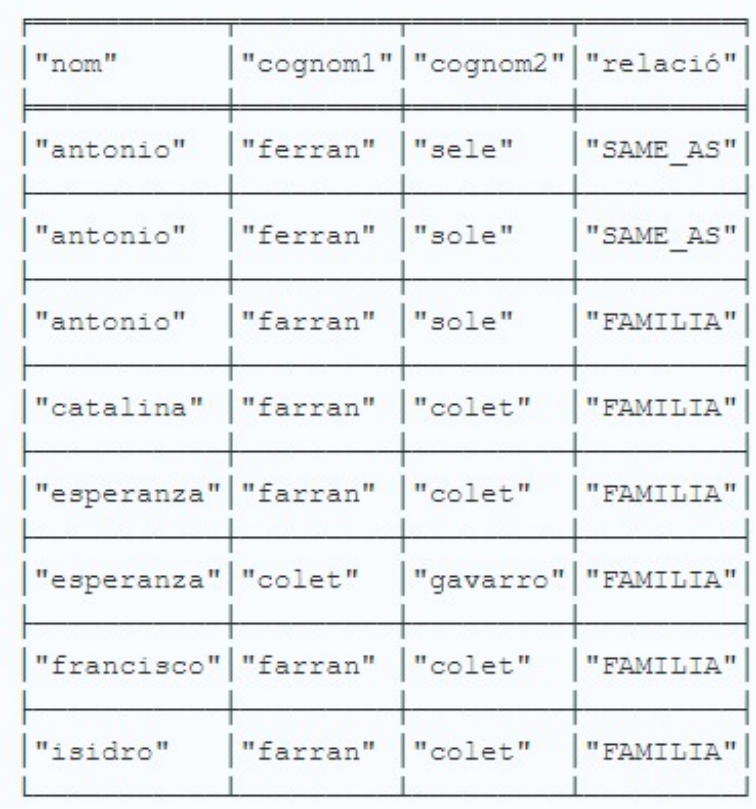
MATCH (i1:Individual)-[rel]-(i2:Individual)

WHERE toLower(i1.name) = 'antonio' and toLower(i1.surname) = 'farran'

RETURN i2.name AS 'nom', i2.surname AS 'cognom1', i2.second_surname AS 'cognom2', type(rel) AS 'relació'

ORDER BY i2.name;

i dona com a resultat



"nom"	"cognom1"	"cognom2"	"relació"
"antonio"	"ferran"	"sele"	"SAME_AS"
"antonio"	"ferran"	"sole"	"SAME_AS"
"antonio"	"farran"	"sole"	"FAMILIA"
"catalina"	"farran"	"colet"	"FAMILIA"
"esperanza"	"farran"	"colet"	"FAMILIA"
"esperanza"	"colet"	"gavarro"	"FAMILIA"
"francisco"	"farran"	"colet"	"FAMILIA"
"isidro"	"farran"	"colet"	"FAMILIA"

Figura 6: Resultat de la cinquena consulta.

2.2.6 *Llisteu totes les relacions familiars que hi ha.*

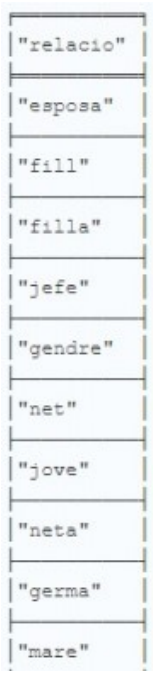
Resolem la consulta buscant totes les relacions entre individus, i ens quedem amb aquelles que tenen la variable *relacio* diferent de *null*. La consulta és

MATCH (a:Individual)-[rel]->(b:Individual)

WHERE rel.relacio <> "null"

RETURN DISTINCT rel.relacio AS relacio;

i dona com a resultat



"relacio"
"esposa"
"fill"
"filla"
"jefe"
"gendre"
"net"
"jove"
"neta"
"germa"
"mare"

Figura 7: Resultat de la sisena consulta.

2.2.7 ***Identifiqueu els nodes que representen el mateix habitatge (carrer i numero) al llarg dels anys de Sant Feliu del Llobregat (SFLL). Mostreu el resultat dels habitatges que tingueu totes dues informacions (carrer i numero), el nombre total d'habitatges, el llistat d'anys dels padrons i el llistat de les Ids de les llars. Ordeneu de més a menys segons el total d'habitatges i mostreu-ne els 10 primers.***

Resolem la consulta agafant tots els habitatges i filtrant per aquells que són de Sant Feliu de Llobregat i no tenen ni carrer ni número nuls. Aleshores retornem de manera agrupada les dades que se'ns demanen. La consulta és

MATCH (a:Habitatge)

WHERE a.municipi = "SFLL"AND a.carrer <> "null"AND a.numero <> "null"

RETURN a.carrer AS carrer, a.numero AS numero, size(collect(a)) AS total, collect(a.any_padro) AS anys, collect(a.id_llar) AS IDs ORDER BY total DESC LIMIT 10;

i dona com a resultat

"name"	"surname"	"second_surname"	"num"
"pablo"	"astruch"	"julia"	7
"jose"	"olle"	"domenech"	6
"benito"	"julivert"	"parera"	6
"jose"	"canals"	"olle"	6
"pedro"	"bargallo"	"ilegible"	6
"jose"	"canals"	"mila"	6
"jose"	"rafuls"	"mila"	5
"jaime"	"jarrey"	"ilegible"	5
"pablo"	"bargallo"	"armangol"	5
"francisco"	"aregay"	"rigol"	5

Figura 8: Resultat de la setena consulta.

2.2.8 *Mostreu les famílies de Castellví de Rosanes amb més de 3 fills. Mostreu el nom i cognoms del cap de família i el nombre de fills. Ordeneu-les pel nombre de fills fins a un límit de 20, de més a menys.*

Resolem la consulta prenent la relació de família entre individus per obtenir-ne els fills. Després, ho relacionem també amb l'habitatge, ja que necessitem el municipi de Castellví de Rosanes. Aleshores filtrem el municipi i, a les relacions, busquem totes aquelles que comencen per "f" menys "familiar". Finalment, en fem un recompte i només retornem aquelles famílies que tinguin més de tres fills. La consulta és

MATCH (a:Habitatge)<-[:VIU]-(b:Individual)<-[rel:FAMILIA]-(c:Individual)

WHERE a.municipi = "CR" AND rel.relacio <> "familiar" AND rel.relacio = "f.*"

WITH c, size(collect(b.id)) AS num

WHERE num > 3

RETURN c.name AS name, c.surname AS surname, c.second_surname AS second_surname, num ORDER BY num DESC LIMIT 20

i dona com a resultat

"name"	"surname"	"second_surname"	"num"
"pablo"	"astruch"	"julia"	7
"jose"	"olle"	"domenech"	6
"benito"	"julivert"	"parera"	6
"jose"	"canals"	"olle"	6
"pedro"	"bargallo"	"ilegible"	6
"jose"	"canals"	"mila"	6
"jose"	"rafuls"	"mila"	5
"jaime"	"jarrey"	"ilegible"	5
"pablo"	"bargallo"	"armangol"	5

Figura 9: Resultat de la vuitena consulta.

2.2.9 *Mitja de fills a Sant Feliu del Llobregat l'any 1881 per família. Mostreu el total de fills, el nombre d'habitatges i la mitja.*

Resolem la consulta agafant la relació entre individus que són família i viuen al mateix habitatge. D'aquests filtrem per any, municipi i que la relació sigui fill o filla. La consulta és

MATCH (a:Individual)-[rel:FAMILIA]->(b:Individual)-[:VIU]->(c:Habitatge)<-[:VIU]-(a)

WHERE c.any_padro = 1881 AND c.municipi = 'SFLL' AND rel.relacio <> "familiar" AND rel.relacio = "f.*"

RETURN count(distinct a) as total_fills, count(distinct c) as num_llars,

round(count(distinct a)/toFloat(count(distinct c)), 2) as mitjana;

i dona com a resultat

"total_fills"	"num_llars"	"mitjana"
1239	580	2.14

Figura 10: Resultat de la novena consulta.

2.2.10 *Per cada any que hi ha a la base de dades, quin és el carrer amb menys habitants de Sant Feliu de Llobregat?*

Resolem la consulta filtrant per municipi i després agrupant les dades fent servir dues clàusules *WITH*. Finalment retornem, per cada any, el carrer amb menys habitants. La consulta és

MATCH (a:Individual)-[:VIU]->(b:Habitatge)

WHERE b.municipi = "SFLL"

WITH b.any_padro as any, b.carrer as carrer, count(a) as total

ORDER BY total

WITH any, collect(carrer)[0] as min_carrer

RETURN any, min_carrer

ORDER BY any;

i dona com a resultat

"any"	"min_carrer"
1833	"carrtera de la part de molins de rey"
1838	"carretera de barna"
1839	"d"
1878	"carrretera"
1881	"s antonio"
1889	"Carretera"

Figura 11: Resultat de la desena consulta.

2.3 Exercici 3. Analítica de grafs

2.3.1 Estudi de les components connexes

El primer que hem fet és fer una projecció del nostre graf principal tenint en compte només les relacions *VIU*, ja que hem considerat oportú dur a terme un estudi sobre quanta gent viu a cada municipi, a cada habitatge per així poder veure també si hi ha gent sense habitatge.

La projecció l'hem feta mitjançant la comanda de cypher

CALL gds.graph.project('Ex3_1', ['Individual', 'Habitatge'], 'VIU');

Inicialment, hem volgut comptar el nombre de components connexes per cada mida de graf. És a dir, el nombre de components connexes amb n nodes per $n = 1, 2, \ldots, 37$. Això és interessant per veure com de connectat està el graf. La comanda que hem fet servir és

CALL gds.wcc.stream('Ex3_1')

YIELD componentId, nodeId

WITH gds.util.asNode(nodeId) AS n1, componentId AS c

WITH size(collect(n1)) AS Components, c

RETURN Components, size(collect(Components)) AS ‘freq‘

ORDER BY Components DESC;

i ens ha retornat

"Components"	"freq"
37	2
36	2
35	5
34	4
33	4
32	9
31	7
30	10
29	20
28	19

Figura 12: Part del resultat de la consulta.

Per visualitzar-ho millor, hem dut a terme un gràfic de barres amb python per poder interpretar millor el resultat obtingut amb cypher, resultant en el següent gràfic.

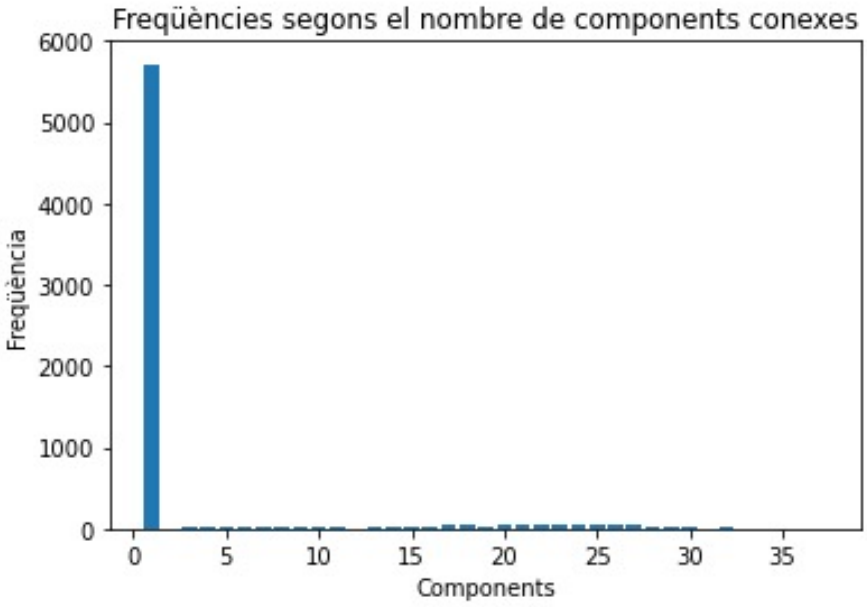


Figura 13: Gràfic de barres.

Clarament, hi ha moltes més components connexes d'un node que de la resta. Podem veure millor el nombre de components connexes amb un cert nombre de nodes diferent a 1 eliminant la barra corresponent a l'1. El resultat és

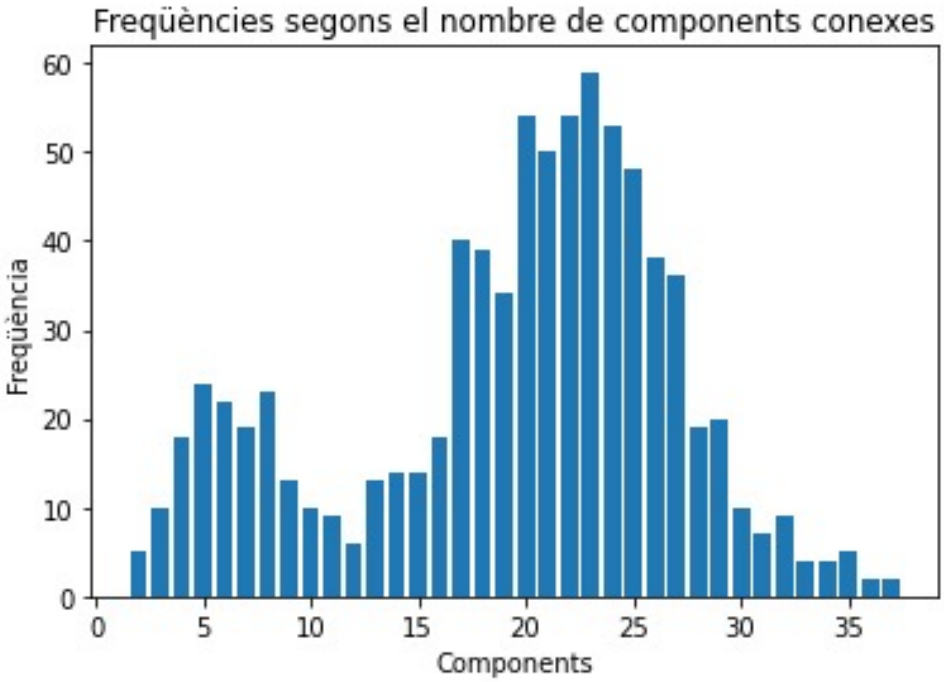


Figura 14: Gràfic de barres sense l'1.

A continuació, hem decidit trobar el nombre de parelles *(Individu)-(Habitatge)* per cada municipi i

any, és a dir, la població de la qual tenim registre d'un municipi en un cert any. Això ho fem amb la comanda

CALL gds.wcc.stream('Ex3_1')

YIELD componentId, nodeId as ind

WITH gds.util.asNode(ind) AS n1, componentId

WITH collect(n1) AS AllNodes, componentId, n1

WHERE n1.municipi <> 'null'

RETURN DISTINCT n1.any_padro AS 'Any_padro', n1.municipi AS 'Municipi',

max(componentId) AS 'Components Connexes';

i ens ha retornat

"Any_padro"	"Municipi"	"Compnent Conexa"
1866	"CR"	341
1881	"SFLL"	8809
1878	"SFLL"	8809
1889	"SFLL"	8809
1838	"SFLL"	4346
1833	"SFLL"	4057
1839	"SFLL"	6289

Figura 15: Resultat de la consulta.

Finalment, hem volgut veure els nodes individu sense habitatge assignat, i ho hem fet amb la consulta

```
CALL gds.wcc.stream('Ex3_1')
YIELD componentId, nodeId
WITH componentId AS c, collect(nodeId) as n1,
size(collect(nodeId)) as components
WHERE components = 1
MATCH (n)
WHERE id(n) in n1
RETURN n;
```

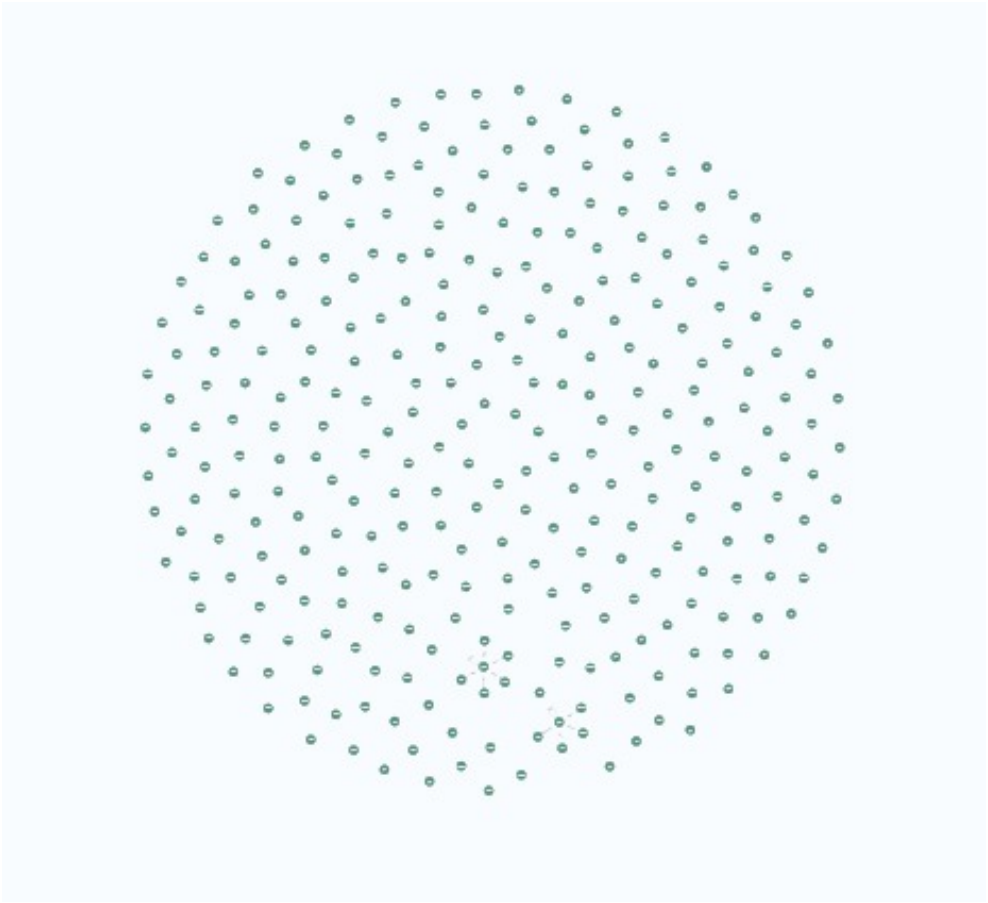


Figura 16: Resultat de la consulta.

També ho hem volgut veure en forma de recompte amb la consulta

```
CALL gds.wcc.stream('Ex3_1')
YIELD componentId, nodeId
WITH componentId AS c, collect(nodeId) as n1,
size(collect(nodeId)) as components
WHERE components = 1
```

MATCH (n)

WHERE id(n) in n1

RETURN count(n) AS Nodes, collect(n.name+“ “+n.surname) AS Persones;

que dona com a resultat

Figura 17: Resultat de la consulta.

2.3.2 Estudi de la semblança entre nodes

Per estudiar la semblança entre nodes, primer ens hem de crear les relacions *MATEIX_HAB*, que uneix els habitatges amb ells mateixos en un altre instant de temps. Això ho fem mitjançant

MATCH (h:Habitatge),(h2:Habitatge)

WHERE h.id_llar <> h2.id_llar

AND h.numero=h2.numero

AND h.municipi=h2.municipi

AND h.carrer=h2.carrer

AND h2.any_padro < h.any_padro

MERGE (h)-[:MATEIX_HAB]->(h2)

la qual cosa ens crea 1745 relacions noves.

A continuació, ens hem creat una projecció que consti dels nodes *Individu* i *Habitatge* amb les relacions *VIU*, *FAMILIA* i *MATEIX_HABITATGE* amb la comanda

CALL gds.graph.project('Ex3_2',['Individual', 'Habitatge'], ['VIU', 'FAMILIA', 'MATEIX_HAB'])

que dona com a resultat

Figura 18: Resultat de la consulta.

Finalment, hem calculat la similaritat entre els nodes de l'últim graf i hem creat les relacions *SIMILAR* en aquells casos on la similaritat superi el 0.45. Això ho hem fet amb la comanda

CALL gds.nodeSimilarity.write('Ex3_2',{

writeRelationshipType:'SIMILAR',

writeProperty:'score',

similarityCutoff:0.45,

topK:5

})

YIELD nodesCompared, relationshipsWritten

A l'executar-la, dona com a resultat la següent imatge.

nodesCompared	relationshipsWritten
14410	59812

Figura 19: Resultat de la consulta.

D'aquí podem veure que, com que hi ha 14410 nodes que són similars a algun altre i 59812 relacions de similaritat, de tal manera que podem dir que, de mitjana, cada node d'aquests 14410 té $\frac{59812}{14410} = 4.15 \approx 4$ relacions de similaritat.

Després, hem trobat habitatges similars i individus similars fent servir

Per comprovar si realment hem detectat similaritats, pel cas dels individus podem mirar quines relacions *SIMILAR* s'han creat entre individus que tenen relació *SAME_AS*. Ho hem fet amb la comanda

MATCH (i1:Individual)-[:SAME_AS]-(i2:Individual)

WHERE (i1)-[:SIMILAR]-(i2)

RETURN i1, i2;

Una part del resultat es veu a la imatge següent.

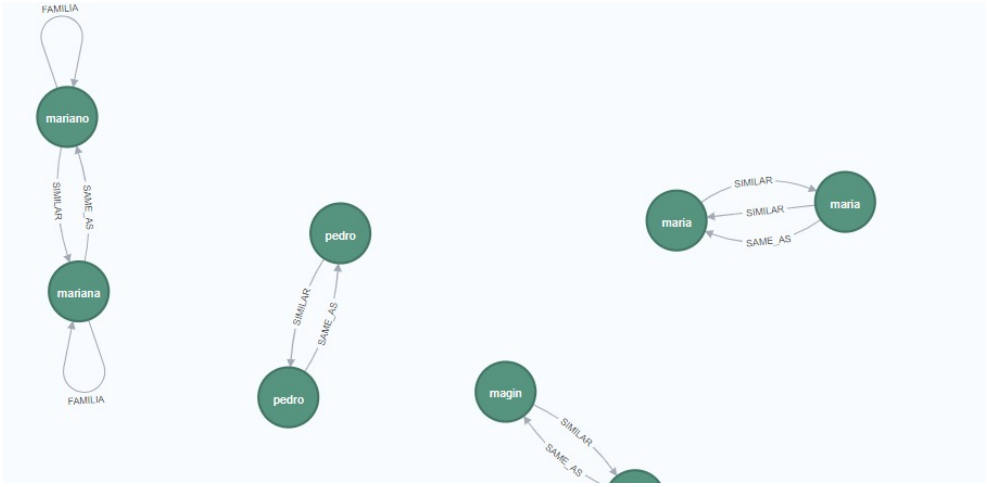


Figura 20: Part del resultat de la consulta.

El que podem veure és que realment la majoria de les relacions són perfectament similars, és a dir, la score és 1.

Hem volgut fer quelcom similar pels habitatges. En aquest cas, com és lògic, hem fet servir la relació *MATEIX_HAB*. A més, però, el que hem fet és comptar en quins casos es crea una relació de similaritat entre dos habitatges que ja estaven relacionats amb *MATEIX_HAB*. Ho hem fet amb la següent comanda.

MATCH (h:Habitatge)-[:MATEIX_HAB]-(h2:Habitatge)

WHERE (h)-[:SIMILAR]-(h2)

RETURN count(*) as Coincidencies;

i ens retorna

Figura 21: Resultat de la consulta.

és a dir, hi ha 744 casos on es detecta una relació de similaritat quan l'habitatge és el mateix. Com que havíem creat 1745 relacions de *MATEIX_HAB*, podem dir que la creació de la similaritat entre habitatges té un $\frac{744}{1745} = 42.63\% \approx 43\%$

3 Repartiment de tasques

A l'iniciar la pràctica, vam repartir les tasques equitativament. L'exercici 1, la importació de dades, la va fer el Sergi. L'exercici 2, que correspon a les consultes de la base de dades, el van fer entre el Marc i el Gerard. El Bernat va fer l'exercici 3, corresponent a analítica de grafs. L'informe el va redactar el Sergi amb aportacions de la resta de membres del grup.

No obstant, tot i haver repartit les parts de la pràctica, tots hem estat al cas també de les parts que inicialment no ens corresponien i hem ajudat en allò que ha calgut, independentment de si se'ns havia assignat o no.